IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Wenbo Mao, et al.	) RE: Claim to Priority
Serial No.: 10/729,299	) Group: Unknown
	) Examiner: Unknown
Filed: December 5, 2003	) Our Ref: B-5321 621552-2
For: "DIGITAL MESSAGE SIGNATURE AND ENCRYPTION"	)
	) Date: January 23, 2004

CLAIM TO PRIORITY UNDER 35 U.S.C. 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

[X] Applicants hereby make a right of priority claim under 35 U.S.C. 119 for the benefit of the filing date(s) of the following corresponding foreign application(s):

COUNTRY	FILING DATE	SERIAL NUMBER
GB	10 April 2003	0308305.2

[ ] A certified copy of each of the above-noted patent applications was filed in the Parent U.S. Application No. .

[X] To support applicants' claim, a certified copy of the above-identified foreign patent application is enclosed herewith.

[ ] The priority documents will be forwarded to the Patent Office when required or prior to issuance.

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first-class mail in an envelope addressed to the "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450", on January 23, 2004 by Katherine Nieva.

Katherine Nieva

Respectfully submitted,

Ross A. Schmitt

Ross A. Schmitt
Attorney for Applicant
Reg. No. 42,529

LADAS & PARRY
5670 Wilshire Boulevard
Suite 2100
Los Angeles, CA 90036
Telephone: (323) 934-2300
Telefax: (323) 934-0202

10/729, 299

The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed Stephen Hordley.

Dated 11 December 2003

Patents Form 1/77

Patents Act 1977
(Rule 16)

THE PATENT OFFICE
E
10 APR 2003
NEWPORT

The Patent Office

1/77

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference — 300205022-1 GB

10 APR 2003
11APR03 E799402-1 D01463
P01/7700 0.00-0308305.2

2. Patent application number *(The Patent Office will fill in this part)* — 0308305.2

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Hewlett-Packard Development Company, L.P.
20555 S.H. 249
Houston, TX 77070
USA

Patents ADP number *(if you know it)* 8557886001

If the applicant is a corporate body, give the country/state of its incorporation

4. Title of the invention — Digital Message Encryption and Authentication

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Richard A. Lawrence
Hewlett-Packard Ltd, IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

Patents ADP number *(if you know it)* 7448078001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	20
Claim(s)	2
Abstract	1
Drawing(s)	4 +4

10. If you are also filing any of the following, state how many against each item.

Priority documents	-
Translations of priority documents	-
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	-
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	-
Any other documents *(please specify)*	Fee Sheet

11. I/We request the grant of a patent on the basis of this application.

Signature Richard A. Lawrence Date 10/4/03

12. Name and daytime telephone number of person to contact in the United Kingdom

Meg Joyce Tel: 0117-312-9068

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

DIGITAL MESSAGE ENCRYPTION AND AUTHENTICATION

This invention relates to a method by which a first computing entity signs and encrypts a message data string, m, a method of signing and encrypting a message data string, *m*, a computing entity programmed to be capable of carrying out the computer implemented steps of such a first computing entity and a computer storage medium having stored thereon a computer program readable by a general-purpose computer, the computer program including instructions for said general purpose computer to configure it to be as such a computing entity

References in square brackets are to the articles listed at Appendix A of this description.

Signcryption is a novel public key primitive first proposed by Zheng in 1997 [14] and as described in US-B-6,396,928. A signcryption scheme combines the functionality of a digital signature scheme with that of an encryption scheme. It therefore offers the three services: privacy, authenticity and non-repudiation. Since these services are frequently required simultaneously, Zheng proposed signcryption as a means to offer them in a more efficient manner that a straightforward composition of digital signature scheme and encryption scheme.

It is only recently that research has been done on defining security for signcryption and providing security arguments for schemes [2,3]. In [3] a scheme similar to the original one proposed in [14] is analysed. The model in [2] is slightly different: it aims to analyse any primitive that achieves the combined functionalility of signature and encryption.

The present invention relates to provably secure signcryption scheme and, in particular, a signcryption scheme based on the RSA trapdoor one-way function.

The present invention in a first aspect is a method by which a first computing entity having an RSA key pair (N_A,e_A), (N_A,d_A) digitally signs and encrypts a message data string, m, for decryption by a second computing entity having an RSA key pair (N_B, e_B), (N_B, d_B), where $|N_A| = |N_B| = n$ and $m \in \{0,1\}^n$, and $k = n + k_0 + k_1$ for integers k_0 and k_1 even, the method comprising:

a) selecting an integer $r \in \{0,1\}^{k_0}$,

b) forming the hash $\omega = H(m \| r)$ where $H : \{0,1\}^{n+k_0} \to \{0,1\}^{k_1}$, and

c) forming the hash $s = G(\omega) \oplus (m \| r)$ where $G : \{0,1\}^{k_1} \to \{0,1\}^{n+k_0}$; steps a) to c) being repeated as necessary to obtain $s \| \omega \leq N_A$; and then

d) signing by forming $c' = (s \| \omega)^{d_A} \bmod N_A$; and, if $c' > N_B$, removing the most significant bit of c' to obtain a new c'; and then

e) encrypting c' by forming $c = c'^{e_B} \bmod N_B$.

In an alternative scheme of the method of the present invention, a first computing entity having an RSA key pair (N_A,e_A), (N_A,d_A) digitally signs and encrypts a message data string, m, for decryption by a second computing entity having an RSA key pair (N_B, e_B), (N_B, d_B), where $|N_A| = |N_B| = n$ and $m \in \{0,1\}^n$, and $k = n + k_0 + k_1$ for integers k_0 and k_1, the method comprising:

a) selecting an integer $r \in \{0,1\}^{k_0}$,

b) forming the hash $\omega = H(m \| r)$ where $H : \{0,1\}^{n+k_0} \to \{0,1\}^{k_1}$, and

c) forming the hash $s = G(\omega) \oplus (m \| r)$ where $G : \{0,1\}^{k_1} \to \{0,1\}^{n+k_0}$, steps a) to c) being repeated as necessary to obtain $s \| \omega \leq N_A$; and then

d) signing by forming $c' = (s \| \omega)^{d_A} \bmod N_A$; steps a) to d) being repeated as necessary to obtain $c' < N_B$; and then

e) encrypting c' by forming $c = c'^{e_B} \bmod N_B$.

The present invention in a further aspect is a computing entity comprising:

a data processing equipment

a memory; and

a communications equipment,

said data processing equipment being configured so as to be capable of processing data according to a set of instructions stored in said memory:

said communications equipment configured so as to communicate data according to said set of instructions;

said set of instructions being such as to configure the computing entity to be capable of carrying out the computer implemented steps of the first computing entity of the methods of the present invention.

In the method of the present invention r may be selected at random

The present invention in a further aspect comprises a computer storage medium having stored thereon a computer program readable by a general-purpose computer, the computer program including instructions for said general purpose computer to configure it to be as the computing entity of the present invention.

An attractive feature of the scheme of the present invention is that if offers non-repudiation in a very simple manner. Non-repudiation for signcryption is not a straightforward sequence of unforgeability like it is for digital signature schemes. The reason for this is that a signcrypted message is “encrypted” as well as “signed”. Therefore, by default, only the intended receiver of a signcryption may verify its authenticity. If a third party is to settle a repudiation dispute over a signcryption, it must have access to some information in addition to the signcryption itself. Of course the receiver could always surrender its private key but this is clearly unsatisfactory. It is often the case that several rounds of zero-knowledge are required. This is not the case for schemes according to the present invention.

The scheme may use a padding scheme similar to PSS [7,8]. The PSS padding scheme was originally designed to create a provably secure signature algorithm when used with RSA [7]. It was subsequently pointed out in [8] that a version of PSS could

also be combined with RSA to create a provably secure encryption function. As demonstrated here, this makes PSS padding perfect for RSA based signcryption. The resulting scheme is very efficient in terms of bandwidth: a signcryption is half the size of a message signed and encrypted using standard techniques for RSA. For this reason we give it the name of Two Birds One Stone. And will be referred to, conveniently, as "TBOS" in this application.

I envisaged that this scheme could be used in an e-commerce scenario such as signcrypting a bankcard payment authorization. Here one RSA block suffices and, as we have discussed, the scheme offers non-repudiation which is clearly desirable for such an application. An alternative use could be signcryption of session keys in a key transport protocol.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings of which:

Figure 1 is a schematic diagram of a system of co-operating computer entities performing the method of the present invention;

Figure 2 is a schematic diagram of the computing entities of the system of Figure 1;

Figure 3 is a high level description of a first embodiment of the method of the present invention;

Figure 4 is a high level description of a second embodiment of the method of the present invention;

In the following description numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without limitation to these specific details. In other instances, well-known methods and structures have not been described in detail so as not to unnecessarily obscure the present invention.

Referring to Figure 1, there is illustrated schematically two computing entities 102, 104, configured for communicating electronic data with each other over a communications network, in this case the internet 106, by communicating data 108,

110, to each other via the internet 106 in well known manner. Illustrated in Figure 1 is first computing entity 02, herein after referred to as entity A or Alice, a second computing entity 104 herein referred to as entity B or Bob. In the example illustrated in Figure 1, the first and second computing entities 102 and 104 are geographically remote from each other and the communications network comprises the known internet 106. In other embodiments and implementations of the present invention the communications network could comprise any suitable means of transmitting digitized data between the computing entities. For example, a known Ethernet network, local area network, wide area network, virtual private circuit or public telecommunications network may form the basis of a communications medium between the computing entities 102 and 104.

The computing entities 102 and 104 have been programmed by storing on memories 203 and 205 programs read from computer program storage media 112 and 114, for example a CD-ROM.

Referring now to Figure 2, there is illustrated schematically physical resources and logical resources of the computing entities A and B. Each computing entity comprises at least one data processing means 200, 202, a memory area 203, 205, a communications port 206, 208, for example, a known Unix operating system. One or more applications programs 22, 214 are configured for operating for receiving, transmitting and performing data processing on electronic data received from other computing entities, and transmitted to other computer entities in accordance with specific methods of the present invention. Optionally there is a user interface 215, 217 which may comprise a visual display device, a pointing device, eg. A mouse or track-ball device, a keypad, and a printer.

Under control of the respective application program 212, 214 each of the computing entities 102, 104 is configured to operate according to a method of the present invention, specific embodiments of which will now be described.

2 Two Birds Ones Stone (TBOS)

2.1 Abstract TBOS

The cryptosystem of the present invention makes use of what will here be called a *permutation with trapdoors*. A permutation with trapdoors $f\colon\{0,1\}^k \rightarrow \{0,1\}^k$ is a function that requires some secret, or "trapdoor", information to evaluate and some different secret information to invert. In the scheme described below it will be assumed that the sender of messages, Alice, knows the secret information necessary to evaluate f, and the receiver, Bob, knows the secret information necessary to evaluate f^{1}.

The scheme may be used to signcrypt messages from $\{0,1\}^n$, where $k = n + k_0 + k_1$ for integers k_0 and k_1. Before f is applied to a message some random padding is applied. The padding used is similar to PSS [7,8]. We describe how the scheme works below.

Parameters.

The scheme requires two hash functions

$$H:\{0,1\}^{n+k_0} \rightarrow \{0,1\}^{k_1} \text{ and } G:\{0,1\}^{k_1} \rightarrow \{0,1\}^{n+k_0}.$$

Signcryption

For Alice to signcrypt a message $m \in \{0,1\}^n$ for Bob:

1. $r \xleftarrow{r} \{0,1\}^{k_0}$
2. $\omega \leftarrow H(m \| r)$
3. $s \leftarrow G(\omega) \oplus (m \| r)$
4. $c \leftarrow f(s \| \omega)$
5. Send c to Bob

Unsigncryption

For Bob to unsigncrypt a cryptogram c from Alice:

1. $s \| \omega \leftarrow f^{-1}(c)$
2. $m \| r \leftarrow G(\omega) \oplus s$
3. If $H(m \| r) = \omega$ accept m

   Else reject

As is stands there is no obvious way to provide non-repudiation. We discuss how this problem is addressed by the present invention in the next section.

2.2 RSA-TBOS

We now show how RSA is used to create something like a permutation with trapdoors, as in Section 2.1, for use with TBOS. It is not claimed that the resulting function is a permutation. This is not necessary for the proof of security.

Referring now to Figure 3, there is shown a pseudo-code flow description of the steps of an embodiment of the present invention by which a first computing entity, "Alice", signcrypts a message,m, for transmittal to a second computing entity, "Bob".

It is assumed sender Alice has generated an RSA *key* pair (N_A,e_A), (N_A,d_A), with $N_A=P_A \cdot Q_A$ *and* $|P_A| = |Q_A| = k/2$. Here and henceforth k is an even positive integer. A receiver Bob is assumed to have done likewise giving him an RSA *key* pair (N_B,e_B), (N_B,d_B). G and H are as described above. Here, if a bit string $\alpha \| \beta$ represents an integer, then α represents the most significant bits of that integer.

Signcryption

For Alice to signcrypt a message $m \in \{0,1\}^n$ for Bob:

1. $r \xleftarrow{r} \{0,1\}^{k_0}$
2. $\omega \leftarrow H(m \| r)$
3. $s \leftarrow G(\omega) \oplus (m \| r)$
4. If $s \| \omega > N_A$ goto 1
5. $c' \leftarrow (s \| \omega)^{d_A} \bmod N_A$
6. If $c' > N_B, c' \leftarrow c' - 2^{k-1}$
7. $c \leftarrow c'^{e_B} \bmod N_B$
8. Send to Bob

Unsigncryption

For Bob to unsigncrypt a cryptogram c from Alice:

1. $c' \leftarrow c^{d_B} \bmod N_B$

2. If $c' > N_A$, reject

3. $\mu \leftarrow c'^{e_A} \bmod N_A$

4. Parse μ as $s \| \omega$

5. $m \| r \leftarrow G(\omega) \oplus s$

6. If $H(m \| r) = \omega$, return m

7. $c' \leftarrow c' + 2^{k-1}$

8. If $c' > N_A$, reject

9. $\mu \leftarrow c'^{e_A} \bmod N_A$

10. Parse μ as $s \| \omega$

11. $m \| r \leftarrow G(\omega) \oplus s$

12. If $\omega \neq H(m \| r)$, reject

13. Return m

The point of step 6 in signcryption is to ensure that $c' < N_B$. If c' initially fails this test then we have $N_A > c' > N_B$. Since both N_A and N_B have k-bits we infer that c' also has k-bits and so the assignment $c' \leftarrow c' - 2^{k-1}$ is equivalent to removing the most significant bit of c'. This gives us $c' < N_B$ as required. Note that this step may cause an additional step in unsigncryption. In particular it may be necessary to perform $c'^{e_A} \bmod N_A$ twice (the two c's will differ by 2^{k-1}). It would have been possible to define an alternative scheme under which the trial and error occurs in signcryption. This would mean repeating steps 1-5 in signcryption with different values of r until $c' < N_B$ is obtained.

Non-repudiation is very simple for RSA-TBOS. The receiver of a signcryption follows the unisgncryption procedure up until stage 2, c' may then be given to a third party who can verify its validity.

3 Security Notions for Signcryption Schemes

3.1 IND-CCA2 for Signcryption Schemes

We take as our starting point the standard definition of indistinguishability of encryptions under adaptive chosen ciphertext attack (IND-CCA2) for public key encryption schemes [1, 4, 5, 10, 11]. A public key encryption scheme enjoys IND-CCA2 security if it is not possible for an adversary to distinguish the encryptions of two messages of its choice under a particular public key, even when it has access to a decryption oracle for this public key. The adversary is able to query the decryption oracle before choosing its two messages and its queries may be determined given information gleaned from previous queries. The adversary is then given the challenge ciphertext i.e. the encryption under the public key in question of one of the two messages chosen at random. It is allowed to continue to query the decryption oracle subject to the condition that it does not query the challenge ciphertext itself. The adversary wins if it correctly guesses which of the two messages was encrypted.

In our definition of IND-CCA2 security for signcryption we allow the adversary access to an unsigncryption oracle for the target receiver's key in a similar manner to that described above for encryption schemes. The difference here is that an oracle for the target receiver's unsigncryption algorithm must be defined with respect to some sender's public key. We therefore consider an attack on two users: a sender and a receiver.

In the case of public key encryption schemes the adversary is able to encrypt any messages that it likes under the public key that it is attacking. This is not the case for signcryption schemes. The private key of the target sender is required in signcryption and so the adversary is not able to produce signcryptions on its own. We must therefore provide the adversary with a signcryption oracle for the keys of the target sender and the target receiver. For an encryption scheme the adversary is able to use its own choice of randomness to generate encryptions, we therefore allow the adversary to choose the randomness used by the signcryption oracle, except for challenge ciphertext generation.

We give a more concrete description of the attack below.

Setup
Using the global systems parameters two private/public key pairs (x_A, Y_A) and (x_B, Y_B) are generated for a target sender/receiver respectively.

Find
The adversary is given Y_A and Y_B, it is also given access to two oracles: a signcryption oracle for Y_A, Y_B and an unsigncryption oracle for Y_A, Y_B. The adversary is allowed to choose the random input as well as the message for the signcryption oracle. At the end of this phase the adversary outputs two messages m_0 and m_1 with $|m_0| = |m_1|$.

Challenge
A bit b is chosen uniformly at random. The message m_b is signcrypted under Y_A, Y_B to produce c^* which is given to the adversary.

Guess
The adversary may continue to query its oracles subject to the condition that it does not query its unsigncryption oracle with c^*. At the end of this phase the adversary outputs a bit b'. The adversary wins if $b' = b$.

If $\mathcal{A}$ is an adversary as described above we define its advantage as:

$$\mathbf{Adv}(\mathcal{A}) = |2 \cdot \Pr[b' = b] - 1|.$$

We say that a signcryption scheme is IND-CCA2 secure if the advantage of any polynomial-time adversary is a negligible[1] function of the security parameter of the scheme.

3.2 Unforgeability of Signcryption Schemes

We adapt the definition of existential unforgeability under adaptive chosen message attack [13] for signature schemes to the signcryption setting.

When using a signature scheme, the only private key used in signature generation belongs to the sender. An adversary can therefore be anyone, since there is no difference in the ability to forge signatures between a receiver of signed messages and a third party. For a signcryption scheme however, signature generation uses the receiver's public key as well as the sender's keys. In this instance there may be a difference in the ability to forge signcryptions between the receiver and a third party, since only the receiver knows the private key corresponding to its public key. With the above in mind we assume that an adversary has access to the private key of the receiver as well as the public key of the sender. It can therefore perform unsigncryption itself.

We allow an adversary to query a signcryption oracle for the target sender's private key. This oracle takes as input a message, and an arbitrary public key chosen by the adversary. The oracle returns the signcryption of the message under the target sender's key and the key chosen by the adversary.

We say that the adversary wins if it produces a valid forged signcryption on some message under the target sender's public key. This message must not have been queried to the signcryption oracle during the attack.

If $\mathcal{A}$ is an adversary as described above we define its advantage as:

$$\mathbf{Adv}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins}].$$

We say that a signcryption scheme is existentially unforgeable under adaptive chosen message attack if the advantage of any polynomial-time adversary is a negligible function of the security parameter of the scheme.

4 IND-CCA2 Security of TBOS

4.1 The Underlying Hard Problem

If the secret information necessary to evaluate a permutation with trapdoors f is made public, then f becomes a standard *trapdoor one-way permutation*. We

[1] A function $\epsilon(k)$ is *negligible* if for every c there exists a k_c such that $\epsilon(k) \leq k^{-c}$ for all $k \geq k_c$.

call this the *induced trapdoor one-way permutation* of f. First of all we consider the security of TBOS under the *partial-domain one-wayness* [12] of the induced trapdoor one-way permutation of f. Let us first state formally the definitions that we will use. Below f will be a trapdoor one-way permutation.

Definition 1 (One-wayness). *The function f is (t, ϵ)-partial domain one-way if the success probability of any adversary $\mathcal{A}$ wishing to recover the preimage of $f(s||\omega)$ in time less than t is upper bounded by ϵ. We state this as:*

$$\mathbf{Adv}_f^{ow}(\mathcal{A}) \leq \Pr_{s,\omega}[\mathcal{A}(f(s||\omega)) = s||\omega] < \epsilon.$$

For any f we denote the maximum value of $Adv_f^{ow}(\mathcal{A})$ over all adversaries running for time t as $Adv_f^{ow}(t)$.

Definition 2 (Partial-domain one-wayness). *The function f is (t, ϵ)-partial domain one-way if the success probability of any adversary $\mathcal{A}$ wishing to recover the partial preimage of $f(s||\omega)$ in time less than t is upper bounded by ϵ. We state this as:*

$$\mathbf{Adv}_f^{pd-ow}(\mathcal{A}) \leq \Pr_{s,\omega}[\mathcal{A}(f(s||\omega)) = \omega] < \epsilon.$$

For any f we denote the maximum value of $Adv_f^{pd-ow}(\mathcal{A})$ over all adversaries running for time t as $Adv_f^{pd-ow}(t)$.

Definition 3 (Set partial-domain one-wayness). *The function f is (l, t, ϵ)-set partial domain one-way if the success probability of any adversary $\mathcal{A}$ wishing to output a set of l elements which contains the partial preimage of $f(s||\omega)$ in time less than t is upper bounded by ϵ. We state this as:*

$$\mathbf{Adv}_f^{s-pd-ow}(\mathcal{A}) \leq \Pr_{s,\omega}[\omega \in \mathcal{A}(f(s||\omega))] < \epsilon.$$

For any f and l we denote the maximum value of $Adv_f^{s-pd-ow}(\mathcal{A})$ over all adversaries running for time t as $Adv_f^{s-pd-ow}(l, t)$.

Suppose that an adversary is given c and successfully returns a set of l elements of which one is ω such that $f(s||\omega) = c$ for some s. It is now possible to break the partial-domain one-wayness of f by selecting one of these elements at random. This tells us that

$$\mathsf{Adv}_f^{pd-ow}(t) \geq \mathsf{Adv}_f^{s-pd-ow}(l, t)/l. \tag{1}$$

4.2 IND-CCA2 Security of Abstract TBOS

Theorem 1. *Let $\mathcal{A}$ be an adversary using a CCA2 attack to break TBOS (as defined in Section 2.1). Suppose that $\mathcal{A}$ has advantage ϵ after running for time t, making at most q_g, q_h, q_s and q_u queries to G, H, the signcryption oracle and the unsigncryption oracle respectively. Suppose that TBOS is implemented with*

k-bit permutation with trapdoors f and let f' be the induced trapdoor one-way permutation of f. We have the following

$$\mathbf{Adv}_{f'}^{pd-ow}(t') \geq \frac{1}{q_g + q_h + q_s} \cdot (\epsilon - 2^{-k_0} \cdot (q_h + q_s) - 2^{-k_1} \cdot q_u)$$

where $t' = t_g \cdot (q_g + q_h + q_s) + t_h \cdot (q_h + q_s) + t_s \cdot q_s + t_u \cdot q_u$, q_g is the time taken to simulate the random oracle G (in the proof of Lemma 1 below) and t_h, t_s and t_u are defined analogously.

This follows from (1) and the following lemma.

Lemma 1. *Using the notation of Theorem 1 we have*

$$\mathbf{Adv}_{f'}^{s-pd-ow}(q_g + q_h + q_s, t') \geq \epsilon - 2^{-k_0} \cdot (q_h + q_s) - 2^{-k_1} \cdot q_u.$$

Proof. We will show how the adversary $\mathcal{A}$ may be used to break the set-partial domain one-wayness of f' by finding the partial preimage of c^* chosen at random from the range of f'. Note that the adversary does not know the secret information necessary to evaluate f. The proof is similar to the corresponding proof in [8].

We will consider an attack on two users Alice, the target sender who knows how to evaluate f, and Bob, the target receiver who knows how to evaluate f^{-1}. We run adversary $\mathcal{A}$ on input of all universal public parameters and the public keys of Alice and Bob. It is necessary to show how to respond to $\mathcal{A}$'s queries to the random oracles G and H and the signcryption/unsigncryption oracles. We denote the algorithms to do this as G_{sim}, H_{sim}, S_{sim} and U_{sim} respectively and we describe them below. To make our simulations sound we keep two lists, L_G and L_H that are initially empty. The list L_G will consist of query/response pairs to the random oracle G. The list L_H will do the same for H. It will also store some extra information as described in H_{sim} below. At the end of the simulation we hope to find the partial preimage of c^* among the queries in L_G.

$G_{sim}(\omega)$
 If $(\omega, x) \in L_G$ for some x:
  Return x
 Else:
  $x \xleftarrow{r} \{0,1\}^{n+k_0}$
  Add (ω, x) to L_G
  Return x

$H_{sim}(m||r)$
 If $(m||r, \omega, c) \in L_H$ for some ω:
  Return ω
 Else:
  $\omega \xleftarrow{r} \{0,1\}^{k_0}$
  $x \leftarrow G_{sim}(\omega)$
  $s \leftarrow x \oplus (m||r)$
  Add $(m||r, \omega, f(s||\omega))$ to L_H
  Return ω

$S_{sim}(m||r)$
 Run $H_{sim}(m||r)$
 Search L_H for entry $(m||r, \omega, c)$
 Return c

$U_{sim}(c)$
 If $(m||r, \omega, c) \in L_H$ for some m:
  Return m
 Else reject

Note that in H_{sim} above we assume that each query has form $m||r$. All this means is each query has length $n + k_0$ bits and so may be parsed as $m||r$ where m has n bits and r has k_0 bits. We make this assumption because, in the random oracle model, it would not help $\mathcal{A}$ to make queries of length different from $n+k_0$.

We also allow $\mathcal{A}$ to make queries of the form $m||r$ to S_{sim} i.e. we allow $\mathcal{A}$ to provide its own random input. This is consistent with a CCA2 attack on an encryption scheme such as RSA-PSS where an adversary can encrypt messages itself using its own random input.

At the end of the find stage $\mathcal{A}$ outputs m_0 and m_1. We choose a bit b uniformly at random and supply the adversary with c^* as the signcryption of m_b. Suppose $c^* = f(s^*||\omega^*)$, this places the following constraints on the random oracles G and H:

$$H(m_b||r^*) = \omega^* \text{ and } G(\omega^*) = s^* \oplus (m_b||r^*). \tag{2}$$

We denote by AskG the event that during $\mathcal{A}$'s attack ω^* has ended up in L_G. We denote by AskH the event the query $m||r^*$ has ended up in L_H for some m.

If $\omega^* \notin L_G$, then $G(\omega^*)$ is undefined and so r^* is a uniformly distributed random variable. Therefore the probability that there exists an m such that $m||r^* \in L_H$ is at most $2^{-k_0} \cdot (q_h + q_s)$. This tells us that

$$\Pr[\mathsf{AskH}|\neg\mathsf{AskG}] \leq 2^{-k_0} \cdot (q_h + q_s). \tag{3}$$

Our simulation U_{sim} can only fail if it outputs reject when it is presented with a valid ciphertext. We denote this event UBad. Suppose that U_{sim} is queried with $c = f(s||\omega)$ and let $m||r = G(\omega) \oplus s$.

We may mistakenly reject a valid ciphertext if $H(m||r) = \omega$, while $m||r$ is not in L_H. Suppose that this query occurs before c^* is given to $\mathcal{A}$ then, since $m||r$ is not in L_H, $H(m||r)$ will take its value at random. If this query is made after c^* is given to $\mathcal{A}$ then $c \neq c^*$ means that $(m, r) \neq (m_b, r^*)$ and so (2) is irrelevant. In either case $H(m||r)$ may take its value at random which means that

$$\Pr[\mathsf{UBad}] \leq 2^{-k_1} \cdot q_u. \tag{4}$$

Let us define the event Bad as

$$\mathsf{Bad} = \mathsf{AskG} \vee \mathsf{AskH} \vee \mathsf{UBad}. \tag{5}$$

Let us denote the event that the adversary wins, i.e. it outputs b' such that $b' = b$, by S. In the event $\neg$Bad the bit b is independent of our simulations, and therefore independent of the adversaries view. We infer from this that

$$\Pr[\mathsf{S}|\neg\mathsf{Bad}] = \frac{1}{2}. \tag{6}$$

Also, in the event $\neg$Bad, the adversary interacts with a perfect simulation of random oracles and signcryption/unsigncryption oracles. This gives us

$$\Pr[\mathsf{S} \wedge \neg\mathsf{Bad}] \geq \frac{1}{2} + \frac{\epsilon}{2} - \Pr[\mathsf{Bad}]. \tag{7}$$

From (6) we obtain

$$\Pr[\mathsf{S} \wedge \neg\mathsf{Bad}] = \Pr[\mathsf{S}|\neg\mathsf{Bad}] \cdot \Pr[\neg\mathsf{Bad}] = \frac{1}{2} \cdot (1 - \Pr[\mathsf{Bad}]). \tag{8}$$

Combining (7) with (8) gives us

$$\Pr[\mathsf{Bad}] \geq \epsilon. \tag{9}$$

From (5) we have

$$\begin{aligned} \Pr[\mathsf{Bad}] &\leq \Pr[\mathsf{AskG} \vee \mathsf{AskH}] + \Pr[\mathsf{UBad}] \\ &= \Pr[\mathsf{AskG}] + \Pr[\mathsf{AskH} \vee \neg\mathsf{AskG}] + \Pr[\mathsf{UBad}] \\ &\leq \Pr[\mathsf{AskG}] + \Pr[\mathsf{AskH}|\neg\mathsf{AskG}] + \Pr[\mathsf{UBad}]. \end{aligned} \tag{10}$$

Together (3), (4) and (10) give us

$$\Pr[\mathsf{AskG}] \geq \epsilon - 2^{-k_0} \cdot (q_h + q_s) - 2^{-k_1} \cdot q_u. \tag{11}$$

The result follows.

4.3 IND-CCA2 Security of RSA-TBOS

We now adapt the result of Section 4.2 to give a proof of the IND-CCA2 security of RSA-TBOS (as defined in Section 2.2) in the random oracle model under the assumption that the RSA function is one-way.

As in Lemma 1 we will assume that there is an adversary $\mathcal{A}$ that runs for time t and has advantage ϵ in breaking the IND-CCA2 security of RSA-TBOS after making at most q_g, q_h, q_s and q_u queries to G, H, the signcryption oracle and the unsigncryption oracle respectively. Given an RSA public key (N_B, e_B), with $N_B = P_B \cdot Q_B$ and $|P_B| = |Q_B| = k/2$, and c^*, we will show how $\mathcal{A}$ may be used to compute the e_B-th root of c^* modulo N_B.

The first step is to generate an RSA key pair (N_A, e_A), (N_A, d_A) with $N_A = P_A \cdot Q_A$ where $|P_A| = |Q_A| = k/2$. We use G_{sim}, S_{sim} and U_{sim} from Lemma 1, we replace H_{sim} with the algorithm below.

$H_{sim}(m||r)$
If $(m||r, \omega, c) \in L_H$ for some ω, return ω
Else:
1. $\omega \xleftarrow{r} \{0,1\}^{k_0}$
2. $x \leftarrow G_{sim}(\omega)$
3. $s \leftarrow x \oplus (m||r)$
4. If $s||\omega > N_A$, goto 1
5. $c' \leftarrow (s||\omega)^{d_A} \mod N_A$
6. If $c' > N_B$, $c' \leftarrow c' - 2^{k-1}$
7. $c \leftarrow c'^{e_B} \mod N_B$
8. Add $(m||r, \omega, c)$ to L_H
9. Return ω

The event Bad is defined as in (5) in the proof of Lemma 1. In our simulation here we are again going to supply $\mathcal{A}$ with c^* as the challenge ciphertext. This

gives us an extra consideration in our simulation. We say that our simulation is Good if (i) $c^{*d_B} \mod N_B < N_A$ and (ii) $gcd(c^{*d_B} \mod N_B, N_A) = 1$. Over the random choices of (N_B, e_B) ,(N_B, d_B), c^* and N_A we have $\Pr[\text{(i)}] = 1/2$ and $\Pr[\text{(ii)}|\text{(i)}] \geq 1 - 2^{-(k/2)+(3/2)}$, hence

$$\Pr[\mathsf{Good}] \geq (2^{-1} - 2^{-\frac{k}{2}+\frac{1}{2}}). \tag{12}$$

Consider (4) in the proof of Lemma 1 for Abstract TBOS. For RSA-TBOS there are two possibilities for a ciphertext to be valid and so we have

$$\Pr[\mathsf{UBad}] \leq 2^{-(k_1-1)} \cdot q_u. \tag{13}$$

We may now use a similar argument as that used to derive (11) in the proof of Lemma 1 to give us

$$\Pr[\mathsf{AskG}|\mathsf{Good}] \geq \epsilon - 2^{-k_0} \cdot (q_h + q_s) - 2^{-(k_1-1)} \cdot q_u \tag{14}$$

in our new simulation. We are interested in the event AskG $\wedge$ Good. We have

$$\Pr[\mathsf{AskG} \wedge \mathsf{Good}] = \Pr[\mathsf{AskG}|\mathsf{Good}] \cdot \Pr[\mathsf{Good}]. \tag{15}$$

Together (12), (14) and (15) tell us

$$\Pr[\mathsf{AskG} \wedge \mathsf{Good}] \geq (2^{-1} - 2^{-\frac{k}{2}+\frac{1}{2}}) \cdot (\epsilon - 2^{-k_0} \cdot (q_h + q_s) - 2^{-(k_1-1)} \cdot q_u) = \delta. \tag{16}$$

Now, in the event AskG $\wedge$ Good we recover a set L_G of size

$$q = q_g + q_h + q_s, \tag{17}$$

containing the k_1 least significant bits of z_0^* where $(z_0^{*d_A} \mod N_A)^{e_B} \mod N_B = c^*$. Call these bits ω_0.

Once we have run our simulation once with challenge ciphertext c^* and obtained L_G we do the following:

For $i = 1, \ldots, \nu - 1$:
  $\alpha_i \xleftarrow{r} \mathbb{Z}^*_{N_B}$
  $c_i^* \xleftarrow{r} c^* \cdot \alpha_i^{e_B} \mod N_B$
  Run the simulation with challenge ciphertext c_i^*
    keeping a list L_{G_i} for G query/response pairs

For $i = 1, \ldots, \nu - 1$ after each run we end up with a list L_{G_i} of size q containing the k_1 least significant bits of $z_0^* \cdot \beta_i \mod N_A$ where $\beta_i = \alpha_i^{e_A} \mod N_A$ with probability at least that of AskG$\wedge$Good as given in (16). Now, if each of the ν runs of our simulation were successful, we have $\omega_0 \in L_G, \omega_1 \in L_{G_1}, \ldots, \omega_{\nu-1} \in L_{G_{\nu-1}}$ such that

$$\begin{aligned} z_0^* &= \omega_0 + 2^{k_1} \cdot x_0 \mod N_A \\ \beta_i \cdot z_0^* &= \omega_i + 2^{k_1} \cdot x_i \mod N_A \text{ for } i = 1, \ldots, \nu - 1 \end{aligned} \tag{18}$$

where z_0^* and $x_0, \ldots, x_\nu$ are unknown. Now, for $i = 1, \ldots, \nu - 1$ let

$$\gamma_i = 2^{-k_1} \cdot (\beta_i \omega_0 - \omega_i) \mod N_A. \tag{19}$$

From (18) and (19) we derive the following for $i = 1, \ldots, \nu - 1$

$$x_i - \beta_i \cdot x_0 = \gamma_i \mod N_A. \tag{20}$$

We have the following lemma from [9].

Lemma 2. *Suppose $2^{k-1} \leq N_A < 2^k$, $k_1 > 64$ and $k/(k_1)^2 \leq 2^{-6}$. If the set of equations (20) has a solution $x = (x_0, \ldots, x_{\nu-1})$ such that $||x||_\infty < 2^{k-k_1}$, then for all values of $\beta = (\beta_1, \ldots, \beta_{\nu-1})$, except for a fraction*

$$\frac{2^{\nu \cdot (k-k_1+\nu+2)}}{N_A^{\nu-1}} \tag{21}$$

of them, this solution is unique and can be computed in time polynomial in ν and in the size of N_A.

It is also shown in [8] that taking $\nu = \lceil (5k)/(4k_1) \rceil$ gives

$$\frac{2^{\nu \cdot (k-k_1+\nu+2)}}{N_A^{\nu-1}} \leq 2^{-k/8}. \tag{22}$$

If we have ν successful runs of our simulation we still do not know which elements of the L_G's form the equations (20) and so to use this method we will have to apply the Lemma 2 algorithm q^ν times. Once we have a solution to (20) we know z_0^* such that $c^* = \left((z_0^{*d_A} \mod N_A)\right)^{e_B} \mod N_B$. From this we may use d_A to compute z^*, the e_B-th root of c^*, as

$$z^* = z_0^{*d_A} \mod N_A. \tag{23}$$

Now, from (16), (20), (22), (23) and Lemma 2 we obtain the result below.

Theorem 2. *Let $\mathcal{A}$ be an adversary that uses a CCA2 attack to attempt to break RSA-TBOS with security parameter k. Suppose that $\mathcal{A}$ succeeds with probability ϵ in time t after making at most q_g, q_h, q_s and q_u queries to G, H, the signcryption oracle and the unsigncryption oracle respectively. In the random oracle model for G and H we may use $\mathcal{A}$ to invert RSA with probability ϵ' in time t' where*

$$\epsilon' \geq \delta^\nu - 2^{-k/8},$$
$$t' \leq \nu \cdot t + (q_g + q_h + q_s)^\nu \cdot poly(k) + 2 \cdot \nu \cdot (q_h + q_s) \cdot T,$$

$\nu = \lceil (5k)/(4k_1) \rceil$, and T is the time it takes for a modular exponentiation.

Note that as is the case in the proofs of security for RSA-OAEP [12], and PSS with standard RSA [8], our reduction is far from tight. Consequently, for the proof of security to be meaningful, we recommend using 2048-bit RSA moduli.

5 Unforgeability of RSA-TBOS

Before we give our security result we must discuss exactly what constitutes a forged RSA-TBOS signcryption. Suppose that we have a user of RSA-TBOS with public key (N_B, e_B). This user can produce a random $c \in \mathbb{Z}_{N_B}^*$ and claim to have forged a signcryption from user who owns key (N_A, e_A). Without knowing (N_B, d_B) it would not be possible to verify this claim. A forged signcryption by the owner of (N_B, d_B) must therefore be presented by following the unsigncryption procedure up until stage 2, c' may then be given to a third party who can verify its validity.

Let us suppose that we have an RSA public key (N_A, e_A) and $c \in \mathbb{Z}_{N_A}^*$ whose e_A-th root we wish to compute. We show in the appendix how to use $\mathcal{A}$, a forging adversary of RSA-TBOS, to do this. This gives the result below.

Theorem 3. *Let $\mathcal{A}$ be an adversary attempting to forge RSA-TBOS signcryptions. Let k be the security parameter of RSA-TBOS. Suppose that $\mathcal{A}$ succeeds with probability ϵ in time t after making at most q_g, q_h and q_s queries to G, H and the signcryption oracle respectively. In the random oracle model we may use $\mathcal{A}$ to invert RSA with probability ϵ' in time t' where*

$$\begin{aligned}
\epsilon' &\geq \epsilon - q_s \cdot \left(2^{-(k_0+1)} \cdot (2q_h + q_s - 1) + 2^{-(k_1+1)} \cdot (2q_g + 2q_h + q_s - 1)\right) \\
&\qquad -2^{-(k_1+1)} \cdot q_h \cdot (2q_g + q_h + 2q_s - 1), \\
t' &\leq t + (q_h + 2q_s) \cdot T,
\end{aligned} \tag{24}$$

where T is the time it takes for a modular exponentiation.

6 Conclusion

We have proposed provably secure signcryption scheme based on the RSA function. This scheme is attractive in that it produces very compact signcryptions with little extra computational cost. Also, our scheme offers non-repudiation in a very simple manner.

In the future it would be interesting to adapt these ideas to produce a scheme that is provably secure under the stronger definitions of security proposed for signcryption in [3]. It is also important to investigate the possibility of a padding scheme for which there exists a tighter security reduction.

Proof of Theorem 2

Our proof technique is similar to one used in [7]. We are going to run the adversary $\mathcal{A}$ in a simulated environment. We first describe or simulation before analysing how it could fail and showing how it could be used to invert the RSA function.

Our simulation must respond $\mathcal{A}$'s queries to the random oracles G and H and the signcryption oracle. We denote the algorithms to do this G_{sim}, H_{sim}, and S_{sim} respectively and we describe them below. To make our simulations sound we keep two lists L_G and L_H that are initially empty. The list L_G will consist of query/response pairs. At the end of the simulation we hope to find the partial preimage of c^* among queries in L_G.

$G_{sim}(\omega)$
 If $(\omega, x) \in L_G$ for some x:
  Return x
 Else:
  $x \xleftarrow{r} \{0,1\}^{n+k_0}$
  Add (ω, x) to L_G
  Return x

$H_{sim}(m||r)$
 If $(m||r, \omega, -) \in L_H$ for some ω:
  Return ω
 Else:
  $x \xleftarrow{r} \mathbb{Z}^*_{N_A}$
  $z \leftarrow x^{e_A} \mod N_A$
  $y \leftarrow c^* z \mod N_A$
  Parse y as $s||\omega$
  Add $(m||r, \omega, x, y, z)$ to L_H
  Add $(\omega, s \oplus (m||r))$ to L_G
  Return ω

$S_{sim}(m||r, (N_B, e_B))$
 $x \xleftarrow{r} \mathbb{Z}^*_{N_A}$
 $y \leftarrow x^{e_A} \mod N_A$
 Parse y as $s||\omega$
 Add $(m||r, \omega, -, -, -)$ to L_H
 Add $(\omega, s \oplus (m||r))$ to L_G
 If $x > N_B$, $x \leftarrow x - 2^{k-1}$
 $c \leftarrow x^{e_B} \mod N_B$
 Return c

Let us now analyse our simulation. Consider events that would cause the adversary's view in our simulated run to differs from it's view in a real attack. Such an event could be caused by an error in G_{sim}, H_{sim} or S_{sim}. We let AskG be the event that there is an error in G_{sim} and define AskH and SBad analogously.

It is easily verified that

$$\Pr[\mathsf{AskG}] = 0. \tag{25}$$

An error in H_{sim} will only occur if it attempts to add $(\omega, s \oplus (m||r))$ to L_G when $G(\omega)$ is already defined. We conclude that

$$\begin{aligned} \Pr[\mathsf{AskH}] &\le 2^{-k_1} \cdot \sum_{i=0}^{q_h - 1} (q_g + q_s + i) \\ &= 2^{-(k_1+1)} \cdot q_h \cdot (2q_g + q_h + 2q_s - 1). \end{aligned} \tag{26}$$

An error in S_{sim} will occur if it attempts to add $(m\|r, \omega, -, -, -)$ to L_H when $H(m\|r)$ is already defined. The only other possibility for an error in S_{sim} is attempting to add $(\omega, s \oplus (m\|r))$ to L_G when $G(\omega)$ is already defined. We conclude that

$$\begin{aligned}\Pr[\mathsf{SBad}] &\leq 2^{-k_0} \cdot \Big(\sum_{i=0}^{q_s-1} (q_h + i) \Big) + 2^{-k_1} \cdot \Big(\sum_{i=0}^{q_s-1} (q_g + q_h + i) \Big) \\ &= q_s \cdot \Big(2^{-(k_0+1)} \cdot (2q_h + q_s - 1) + 2^{-(k_1+1)} \cdot (2q_g + 2q_h + q_s - 1) \Big).\end{aligned} \tag{27}$$

We also define the event FBad to be that when $\mathcal{A}$ outputs a valid forged signcryption c on some message m, but $m\|r$ was never a query to H_{sim}. Clearly we have

$$\Pr[\mathsf{FBad}] \leq 2^{-k_1}. \tag{28}$$

We define the event Bad to be

$$\mathsf{Bad} = \mathsf{AskG} \vee \mathsf{AskH} \vee \mathsf{SBad} \vee \mathsf{FBad}. \tag{29}$$

Let us consider the event $\mathcal{A}$ wins $\wedge \neg$Bad in our simulated run of $\mathcal{A}$. If this event occurs then $\mathcal{A}$ outputs a forged signcryption c of some m such that $(m\|r, \omega, x, y, z) \in L_H$ for some r, ω, x, y, z. Now, looking at the construction of H_{sim} we see that we have

$$(c/x)^{e_A} = (y/x^{e_A}) = (y/z) = (c^* z/z) = c^* \mod N_A. \tag{30}$$

Therefore $(c/x) \mod N_A$ is the e_A-th root of c^* modulo N_A as required. We denote the event that we manage to find the e_A-th root modulo N_A of c^* by Invert. We see from (30) that

$$\Pr[\mathsf{Invert}]_{sim} \geq \Pr[\mathcal{A} \text{ wins} \wedge \neg\mathsf{Bad}]_{sim}, \tag{31}$$

where the subscript *sim* denotes the fact that these are probabilities in our simulated run of $\mathcal{A}$. We will denote probabilities in a real execution of $\mathcal{A}$ with the subscript *real*. From (31) and the definition of Bad we see that

$$\Pr[\mathsf{Invert}]_{sim} \geq \Pr[\mathcal{A} \text{ wins} \wedge \neg\mathsf{Bad}]_{real} \geq \Pr[\mathcal{A} \text{ wins}]_{real} - \Pr[\mathsf{Bad}]_{real}. \tag{32}$$

The result now follows from (25), (26), (27), (28), (29) and (32).

Appendix A

References

1. M. Abdalla, M. Bellare and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In Topics in Cryptology - CT-RSA 2001, volume 2020 of Lecture Notes in Computer Science, pages 143-158. Springer-Verlag, 2001.
2. J. H. An and Y. Dodis and T. Rabin. On the Security of Joint Signature and Encryption, In Advances in Cryptology - EUROCRYPT 2002, volume 2332 of Lecture Notes in Computer Science, pages 83-107. Springer-Verlag, 2002.
3. J. Baek, R. Steinfeld and Y. Zheng. Formal Proofs for the Security of Signcryption. In Public Key Cryptography 2002, volume 2274 of Lecture Notes in Computer Science, pages 80-98. Springer-Verlag, 2002.
4. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In Advances in Cryptology - CRYPTO '98, volume 1462 of Lecture Notes in Computer Science, pages 26-45. Springer-Verlag, 1998.
5. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption - How to Encrypt with RSA. In Advances in Cryptology - EUROCRYPT '94, volume 950 of Lecture Notes in Computer Science, pages 92-111. Springer-Verlag, 1994.
6. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the First ACM Conference on Computer and Communications Security, pages 62-73. 1993.
7. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to sign with RSA and Rabin. In Advances in Cryptology - EUROCRYPT '96, volume 1070 of Lecture Notes in Computer Science, pages 399-416. Springer-Verlag, 1996.
8. J.-S. Coron and M. Joye and D. Naccache and P. Paillier. Universal Padding Schemes for RSA. In Advances in Cryptology - CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science, pages 226-241. Springer-Verlag, 2002.
9. J.-S. Coron and M. Joye and D. Naccache and P. Paillier. Universal Padding Schemes for RSA. Full version from http://eprint.iacr.org/2002/115/. 2002.
10. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In Advances in Cryptology - CRYPTO '98, volume 1462 of Lecture Notes in Computer Science, pages 13-25. Springer-Verlag, 1998.
11. R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. Available at http://eprint.iacr.org/2001/108/, 2001.
12. E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern. RSA-OAEP Is Secure under the RSA Assumption. In Advances in Cryptology - CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, pages 260-274. Springer-Verlag, 2001.
13. S. Goldwasser, S. Micali and R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM Journal on Computing, 17(2):281-308, 1988.
14. Y. Zheng. Digital Signcryption or How to Achieve Cost(Signature & Encryption) $<<$ Cost(Signature) + Cost(Encryption). In Advances in Cryptology - CRYPTO '97, volume 1294 of Lecture Notes in Computer Science, pages 165-179. Springer-Verlag, 1997.

CLAIMS

1. A method by which a first computing entity having an RSA key pair (N_A,e_A), (N_A,d_A) digitally signs and encrypts a message data string, *m*, for decryption by a second computing entity having an RSA key pair (N_B, e_B), (N_B, d_B), where $|N_A| = |N_B| = n$ and $m \in \{0,1\}^n$, and $k = n + k_0 + k_1$ for integers k_0 and k_1 even, the method comprising:

a) selecting an integer $r \in \{0,1\}^{k_0}$,

b) forming the hash $\omega = H(m \| r)$ where $H:\{0,1\}^{n+k_0} \rightarrow \{0,1\}^{k_1}$, and

c) forming the hash $s = G(\omega) \oplus (m \| r)$ where $G:\{0,1\}^{k_1} \rightarrow \{0,1\}^{n+k_0}$; steps a) to c) being repeated as necessary to obtain $s \| \omega \leq N_A$; and then

d) signing by forming $c' = (s \| \omega)^{d_A} \bmod N_A$; and, if $c' > N_B$, removing the most significant bit of c' to obtain a new c'; and then

e) encrypting *c*' by forming $c = c'^{e_B} \bmod N_B$.

2. The method as claimed in claim 1 in which r is selected at random.

3. A method by which a first computing entity having an RSA key pair (N_A,e_A), (N_A,d_A) digitally signs and encrypts a message data string, *m*, for decryption by a second computing entity having an RSA key pair (N_B, e_B), (N_B, d_B), where $|N_A| = |N_B| = n$ and $m \in \{0,1\}^n$, and $k = n + k_0 + k_1$ for integers k_0 and k_1 even, the method comprising:

a) selecting an integer $r \in \{0,1\}^{k_0}$,

b) forming the hash $\omega = H(m \| r)$ where $H:\{0,1\}^{n+k_0} \rightarrow \{0,1\}^{k_1}$, and

c) forming the hash $s = G(\omega) \oplus (m \| r)$ where $G:\{0,1\}^{k_1} \rightarrow \{0,1\}^{n+k_0}$, steps a) to c) being repeated as necessary to obtain $s \| \omega \leq N_A$; and then

d) signing by forming $c' = (s \| \omega)^{d_A} \bmod N_A$; steps a) to d) being repeated as necessary to obtain $c' < N_B$; and then

e) encrypting c' by forming $c = c'^{e_B} \bmod N_B$.

4. The method as claimed in claim 3 in which r is selected at random.

5. A computing entity comprising:

a data processing equipment

a memory; and

a communications equipment,

said data processing equipment being configured so as to be capable of processing data according to a set of instructions stored in said memory:

said communications equipment configured so as to communicate data according to said set of instructions;

said set of instructions being such as to configure the computing entity to be capable of carrying out the computer implemented steps of the first computing entity of any one of claims 1 to 4.

6. A computer storage medium having stored thereon a computer program readable by a general-purpose computer, the computer program including instructions for said general purpose computer to configure it to be as the computing entity of claim 5.

7. A computing entity arranged to digitally sign and encrypt a message substantially as hereinbefore described.

8. A method of digitally signing and encrypting a message substantially as hereinbefore described.

9. A computer storage medium having stored thereon a computer program readable by a general-purpose computer, the computer program including instructions for said general purpose computer to configure it to be as a computing entity substantially as hereinbefore described.

ABSTRACT (Ref Fig 3)

DIGITAL MESSAGE ENCRYPTION AND AUTHENTICATION

A method by which a first computing entity having an RSA key pair (N_A,e_A), (N_A,d_A) digitally signs and encrypts a message data string, m, for decryption by a second computing entity having an RSA key pair (N_B, e_B), (N_B, d_B), where $|N_A| = |N_B| = n$ and $m \in \{0,1\}^n$, and $k = n + k_0 + k_1$ for integers k_0 and k_1 even, the method comprising:

a) selecting an integer $r \in \{0,1\}^{k_0}$,

b) forming the hash $\omega = H(m \| r)$ where $H:\{0,1\}^{n+k_0} \to \{0,1\}^{k_1}$, and

c) forming the hash $s = G(\omega) \oplus (m \| r)$ where $G:\{0,1\}^{k_1} \to \{0,1\}^{n+k_0}$; steps a) to c) being repeated as necessary to obtain $s \| \omega \leq N_A$; and then

d) signing by forming $c' = (s \| \omega)^{d_A} \bmod N_A$; and, if $c' > N_B$, removing the most significant bit of c' to obtain a new c'; and then

e) encrypting c' by forming $c = c'^{e_B} \bmod N_B$.

This signcryption scheme based on RSA and is proven secure in the random oracle model [6] for its privacy and unforgeability. The proofs are under the assumption that inverting the RSA function is hard. The scheme produces compact ciphertexts as well as offering non-repudiation in a very straightforward manner.

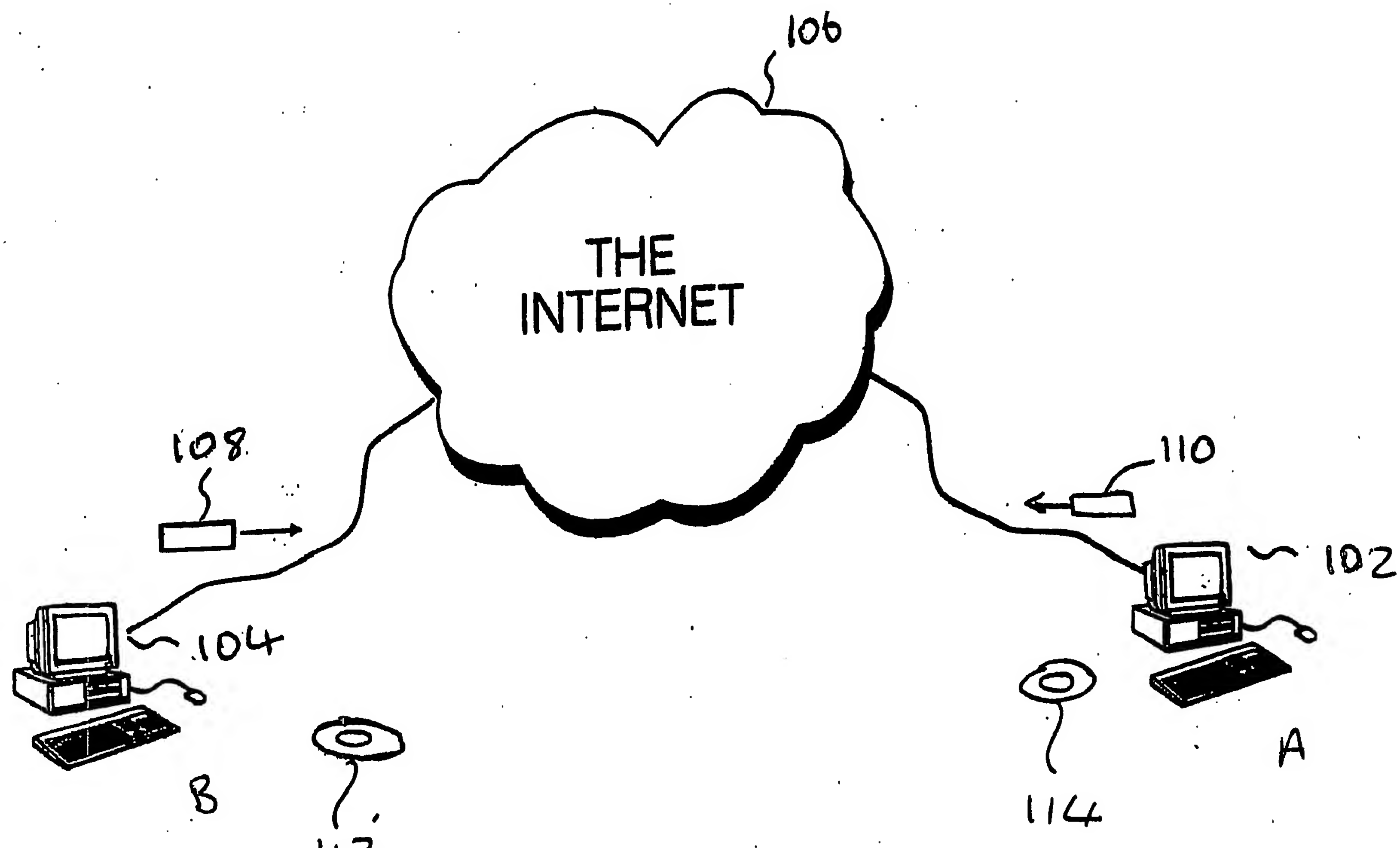


Fig 1

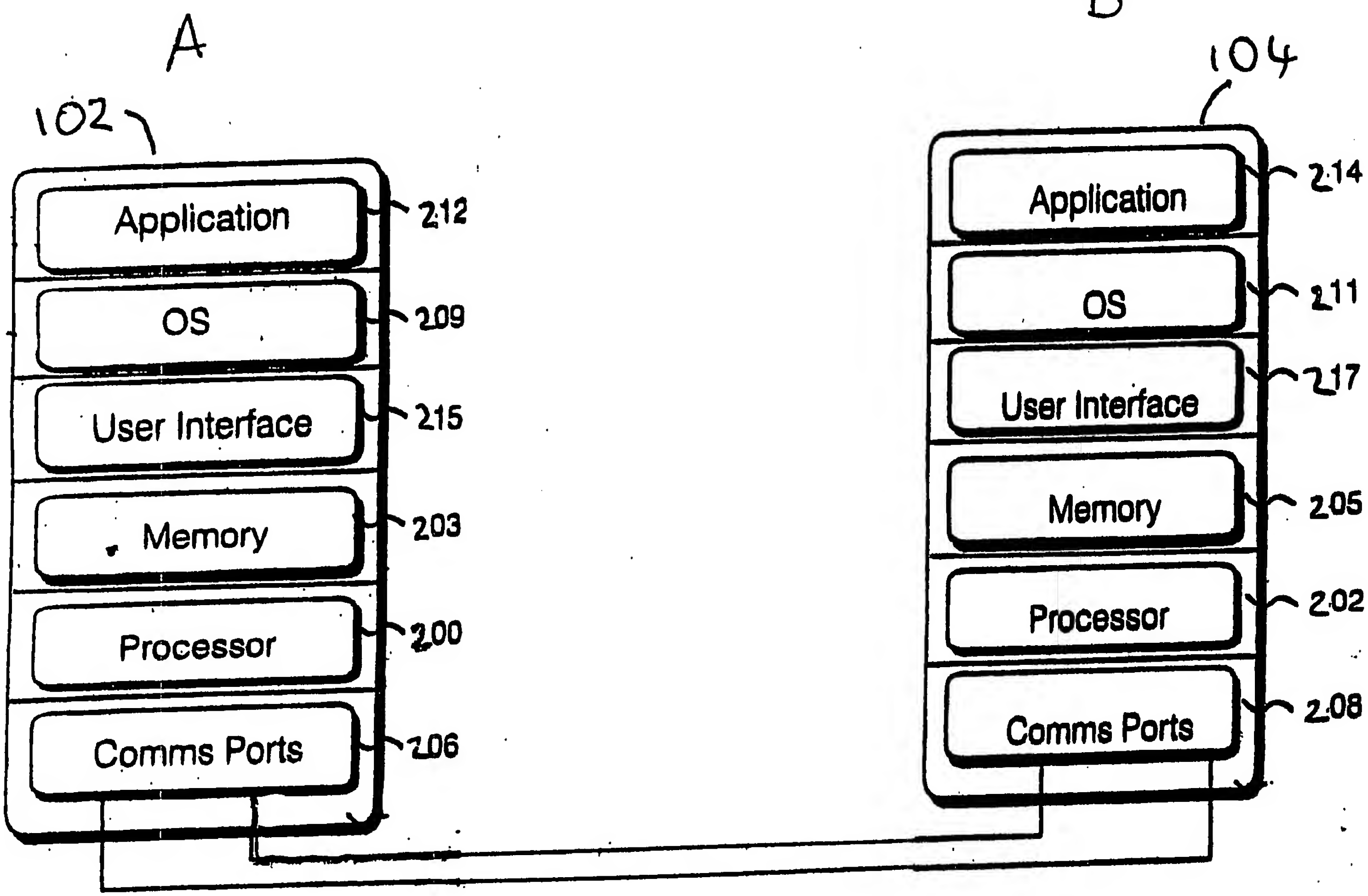
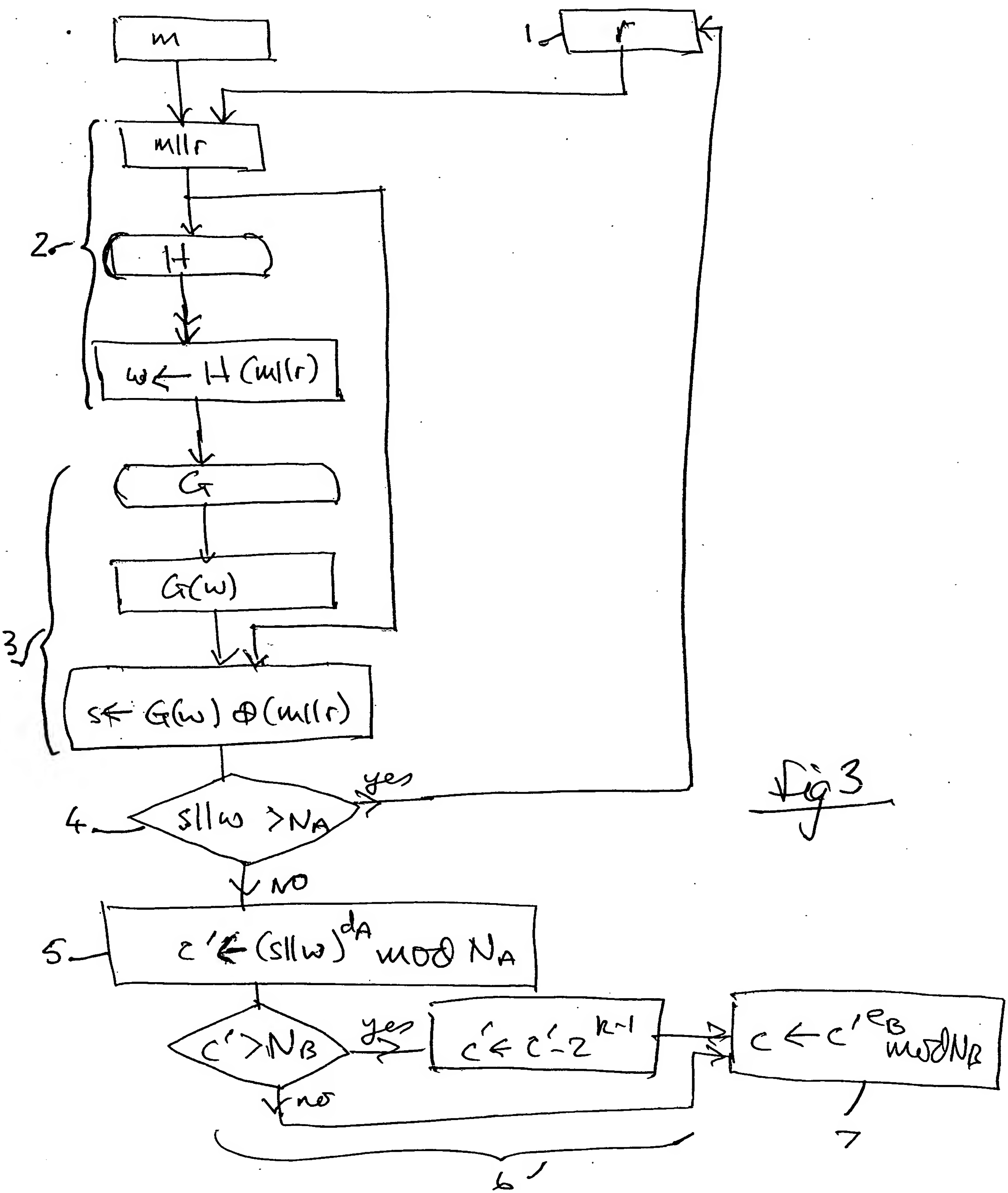

A
102
Application
OS
User Interface
Memory
Processor
Comms Ports
212
209
215
203
200
206
B
104
Application
OS
User Interface
Memory
Processor
Comms Ports
214
211
217
205
202
208

Fig 2



m
r
1
m||r
2
H
w ← H(m||r)
G
G(w)
3
s ← G(w) ⊕ (m||r)
s||w > N_A
yes
4
NO
5
c' ← (s||w)^dA mod N_A
c' > N_B
yes
c' ← c' − 2^(k−1)
c ← c'^eB mod N_B
no
6
7

Fig 3

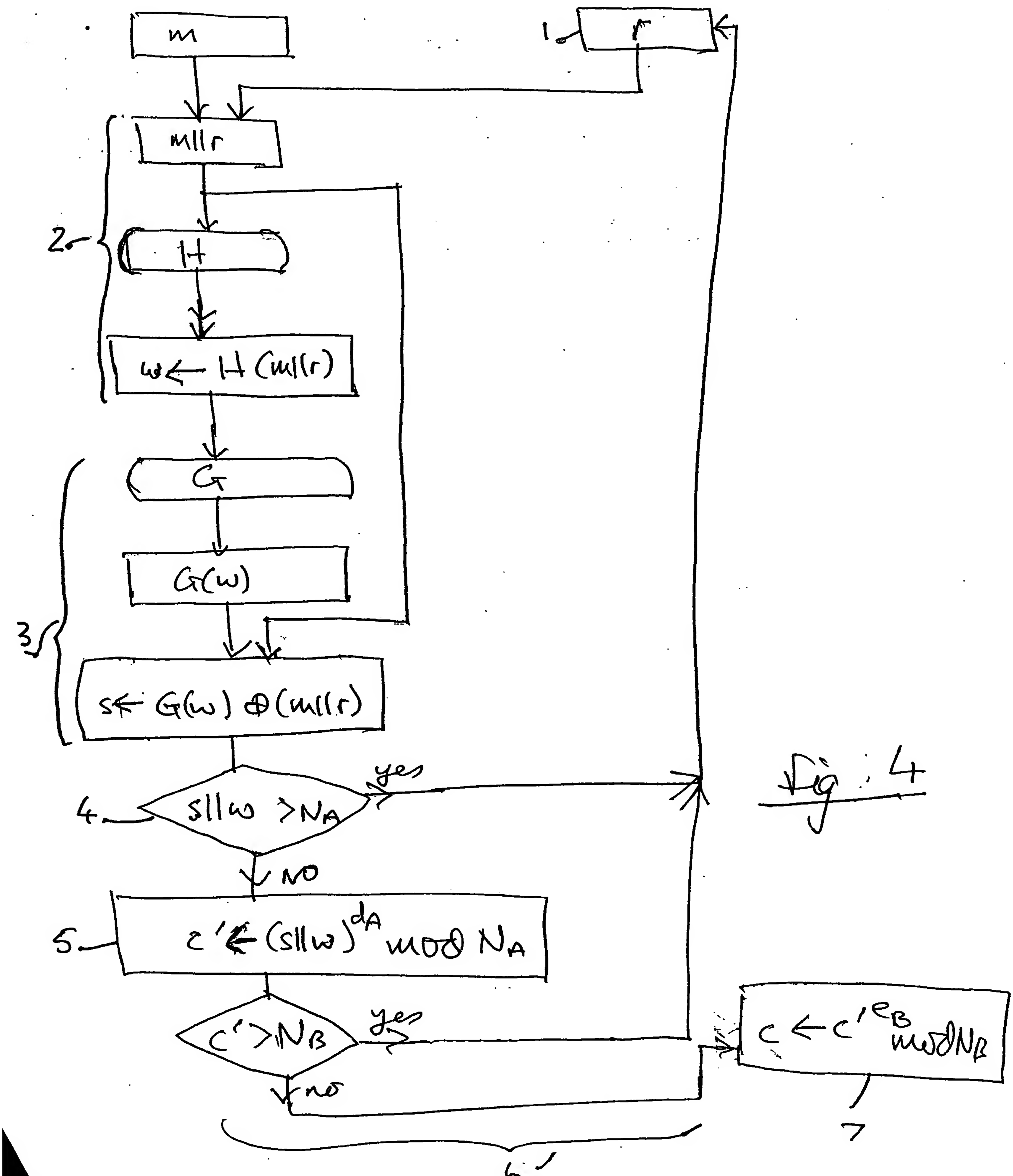

m
1
r
m||r
2
H
w ← H(m||r)
G
G(w)
3
s ← G(w) ⊕ (m||r)
4
s||w > N_A
yes
NO
5
c' ← (s||w)^{d_A} mod N_A
c' > N_B
yes
no
c ← c'^{e_B} mod N_B
7
6

Fig. 4